Communication Cost in Parallel Query Evaluation A Tutorial

Dan Suciu
University of Washington

We consider the following problem: what is the amount of communication required to compute a query in parallel on p servers, over a large input database? To study this problem we define a variant of Valiant's BSP model [10], called the Massively Parallel Communication (MPC) model, where servers are infinitely powerful and where the cost is measured in terms of the maximum communication per server, and the number of rounds. Query evaluation in this model has been studied for full conjunctive queries in [6, 7, 9]. The model is similar to the MapReduce model of computation, where full conjunctive queries were studied in [1–4].

This tutorial presents parallel algorithms for conjunctive queries and proves lower bounds, under various settings. Using Atserias, Grohe, and Marx' (AGM) upper bound on the query size [5] one can derive a lower bound for the MPC model expressed in terms of the fractional edge covering number of the query's hypergraph, however, no matching algorithm is known for this bound. All algorithms are based on Afrati and Ullman's *Shares* algorithm for the MapReduce model [4], called the *HyperCube* algorithm for the MPC model [6]. The algorithm is based on partitioning the input data using randomized hash functions, and requires the data to be skew-free to ensure uniform partitioning.

We will start by discussing the case when the computation is limited to a single round of communication. In this case, if the data is skew-fee, then a tight bound is given in terms of the fractional vertex (not edge!) covering number of the query's hypergraph, and this result can be extended to skewed input data. Next, we consider the multi-round case. Here, the key algorithmic ingredient is a technique that uses additional rounds in order to handle skewed values in the data. Using this technique it has been shown recently [8] that the tight bound for evaluating a query where all relations have arity at most two is given by the general bound derived from the AGM inequality. The case when the input relations have arbitrary arities remains open; it is not known whether this bound is given in terms of fractional edge cover, or the fractional vertex cover.

Supported by NSF AITF-1535565 and IIS-1247469.

PODS'17, May 14-19, 2017, Chicago, IL, USA

DOI: http://dx.doi.org/10.1145/3034786.3056449

1. REFERENCES

[1] F. N. Afrati, M. Joglekar, C. Ré, S. Salihoglu, and J. D. Ullman. GYM: A multiround join algorithm in mapreduce. *CoRR*, abs/1410.4156, 2014.

[2] F. N. Afrati, A. D. Sarma, S. Salihoglu, and J. D. Ullman. Upper and lower bounds on the cost of a map-reduce computation. *PVLDB*, 6(4):277–288, 2013.

[3] F. N. Afrati, N. Stasinopoulos, J. D. Ullman, and A. Vasilakopoulos. Sharesskew: An algorithm to handle skew for joins in mapreduce. *CoRR*, abs/1512.03921, 2015.

[4] F. N. Afrati and J. D. Ullman. Optimizing joins in a map-reduce environment. In *EDBT 2010, 13th International Conference on Extending Database Technology, Lausanne, Switzerland, March 22-26, 2010, Proceedings*, pages 99–110, 2010.

[5] A. Atserias, M. Grohe, and D. Marx. Size bounds and query plans for relational joins. *SIAM J. Comput.*, 42(4):1737–1767, 2013.

[6] P. Beame, P. Koutris, and D. Suciu. Communication steps for parallel query processing. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 273–284, 2013.

[7] P. Beame, P. Koutris, and D. Suciu. Skew in parallel query processing. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 212–223, 2014.

[8] B. Ketsman and D. Suciu. A worst-case optimal multi-round algorithm for parallel computation of conjunctive queries. In *PODS*, 2017.

[9] P. Koutris, P. Beame, and D. Suciu. Worst-case optimal algorithms for parallel query processing. In *19th International Conference on Database Theory, ICDT 2016, Bordeaux, France, March 15-18, 2016*, pages 8:1–8:18, 2016.

[10] L. G. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33(8):103–111, 1990.